基于 FP-Growth 的智能家居用户时序关联操控习惯挖掘方法 *

梁天恺[a]，曾　碧[a]，刘建圻[b]

(广东工业大学 a.计算机学院; b.自动化学院，广州 510006)

摘　要：针对传统关联规则挖掘算法无法高效且准确地挖掘出隐含于用户操作记录中的时序关联操控习惯，提出一种基于 FP-Growth 的智能家居用户时序关联操控习惯挖掘算法。该算法分为三个阶段，分别基于用户操控动作森林、改进的 FP-Growth 算法和一种时间约束规则进行事务集的生成、时序频繁项集的生成以及最终时序关联操控习惯的生成。最后，使用真实用户操控记录进行对比实验，结果表明该算法能提高生成事务集的效率，并能更准确地发现用户操控家居设备的时序关联习惯。

关键词：智能家居；行为预测；数据挖掘；关联分析；个性化推荐

中图分类号：TP301.6　　doi: 10.19734/j.issn.1001-3695.2018.07.0527

FP-Growth-based user temporal association control habits mining method for smart home

Liang Tiankai[a], Zeng Bi[a], Liu Jianqi[b]

(a. *School of Computer, b. School of Automation, Guangdong University of Technology, Guangzhou* 510006, *China*)

Abstract: Concern the problem that the traditional association analysis algorithms cannot efficiently and accurately mine the user's potential temporal association control habits which are implied in the user's operation records, this paper proposed a novel user temporal association control habits mining method based on FP-Growth. This method includes three stages: to generate the transaction set, the temporal frequent item set, and the final temporal association control habits via the user operation-action forest, the improved FP-Growth algorithm and a time constraint rule. Finally, the comparative experiments by using the real user control records show that this method can improve the efficiency of transaction set generation and can more accurately discover the user's temporal association habits of smart home devices.

Key words: smart home; behavior prediction; data mining; association analysis; personalized recommendation

0　引言

随着物联网技术、计算机网络技术以及数据挖掘技术的跨越式发展，智能化成为新世纪发展趋势的新代名词。在此大背景下，智能家居的发展也呈现出突发猛进的态势[1]。智能家居系统是以个人家居空间为主要平台，通过物联网技术将家居设备连接到网络中，实现家居设备的远程操控，构建高效的住宅设施与家庭日程事务的管理系统[2]。智能家居系统的智能化水平分为三层[3]。低级智能化水平只实现智能家居设备的简单远程操作。即用户可以通过移动终端将控制指令通过无线/有线网络发送到智能家居系统的控制中心，然后控制中心通过家居网络将指令下发给相应的家居设备，最终实现智能家居设备的远程操作[4]；中级智能化水平则实现了基于环境感知的家居设备触发式自动化控制[5]。例如，用户自定义规则"当室内温度高于 30 摄氏度时，帮我打开空调"。此后，智能家居系统将通过连入到家居网络的温度传感器来感知室内环境，当室内环境的变化达到触发预定规则的阈值时，即室内温度高于 30 摄氏度时，系统将自动进行相应的操控，即打开空调；最高级别的智能化水平则要求智能家居系统具备学习能力，能从大量用户操控记录中学习到用户对家居设备的操控习惯，并代替用户自主地在适当条件下操控家居设备，实现家居设备真正意义的智能操作[6]。

用户对智能家居设备的各种操控行为之间存在一定的联系和规律，若能将这种潜在的关联性操控习惯从用户的历史操控记录中识别出来，并利用其进一步开发出更了解用户的智能家居系统，有利于智能家居行业的发展更趋向于智能家居系统的最高层次智能化水平[7,8]。

目前，国内外学者对用户的关联性操控习惯的挖掘的研究已有一定的成果。例如，文献[9]提出使用 Apriori 关联分析算法来挖掘隐含在用户历史交互记录中的关联操控习惯。但该方法所得到的关联性操控习惯规则，只是简单表达出用户各操控动作之间的关系，并未能完整表达出用户操控动作之间的时间特性，因此该算法的学习能力比较低下。相似的还有文献[10]提出的一种基于假设检验的关联分析算法。该算法进一步证明家居设备间的确存在较强依赖关系且能可被有效挖掘。其次，为了提高关联分析算法对混杂数据以及缺失数据的抗噪能力，Cook 等人[11]利用情节发现（Episode Discovery，简称 ED）算法来识别用户的操控行为中隐含的关联关系，一定程度上提高了算法处理大规模混杂数据的能力。学者们为了进一步提高算法对大规模数据的处理能力，文献[12]基于 ART（adaptive resonance theory）网络提出一种双层 ART1 模式分类方法，对用户的操控动作之间的关系建立数学模型，同时利用云端仓库对用户操控记录数据进行存储，有效解决了本地存储资源无法适应大规模用户操控数据

收稿日期：2018-07-20；**修回日期：**2018-09-10　**基金项目：**国家自然科学基金青年基金资助项目（61701122）；广东省产学研重大专项资助项目（2016B010108004）；广州市重点科技项目（201604020016）；广东省产学研专项资助项目（2014B090904080）

作者简介：梁天恺（1993-），男，广东肇庆人，硕士，主要研究方向为数据挖掘（tiankai.liang.gdut@outlook.com）；曾碧（1963-），女，广东广州人，教授，博士，主要研究方向为智能信息处理，智能机器人；刘建圻（1982-），男，江西赣州人，副教授，博士，主要研究方向为物联网与大数据技术.

的缺点。虽然上述方法对智能家居应用的推广以及智能家居系统的智能化水平的提高具有较大的意义和启发，然而上述的算法依旧无法保证所挖掘到用户关联操控习惯内的各规则子项之间存在时序性以及强烈的时间关联性，无法准确挖掘出用户潜在的时序关联操控习惯。

为解决上述问题，本文提出一种基于 FP-Growth 算法的时序关联规则分析（temporal association analysis based on FP-Growth，简称 TAABFPG）算法。TAABFPG 算法提出使用森林的方式来存储用户操控记录和生成事务集；其次，为了准确挖掘出用户潜在的具有强烈时间关联性的时序关联操控习惯，还提出使用一种简单但有效的约束规则。最后，本文使用由智能家居企业所提供的真实用户数据与 3 种经典且常见的关联分析算法中进行了对比实验，证明本算法的有效性与性能的优越性。

1 算法框架

本文所提出的基于 FP-Growth 的智能家居用户关联操控习惯挖掘（TAABFPG）算法可分为三个阶段：事务集生成、时序频繁项集生成以及最终的用户关联操控习惯的生成，算法流程如图 1 所示，其具体执行步骤如下：

a)事务集的生成：将用户历史操控数据进行动作化处理，并生成用户操控动作森林，最后通过遍历森林得到事务集。

b)时序频繁项集的生成：本文基于 FP-Growth 算法，对 FP 树生成过程添加了时序化处理的改进，使其可以产生满足最小支持度的时序频繁项集。

c)用户关联操控习惯的生成：此步骤包括时序候选关联规则的生成以及最终的用户操控习惯的筛选两部分。首先，依据频繁项集生成满足最小置信度的时序候选关联规则，并计算候选关联规则时间约束因子，进而通过时间约束规则挖掘出用户具有时间约束的时序关联操控习惯。

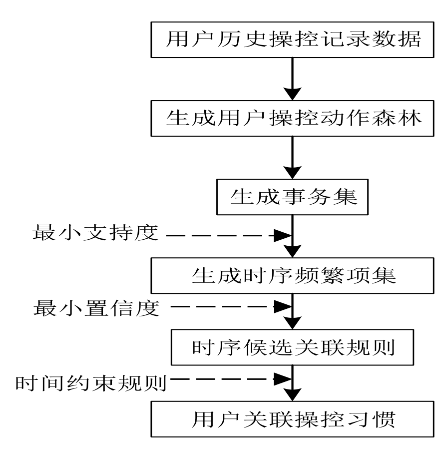


图 1 TAABFPG 算法流程框图

Fig. 1 Schematic diagram of TAABFPG algorithm

TAABFPG 算法流程描述如下：

输入: *min*(*supp*)：规则的最小支持度，

min(*conf*)：规则的最小置信度，

q:规则子项之间的时间约束系数(单位:分钟)

dataset：用户操控记录。

输出：具有时间约束的时序关联操控习惯

a)根据用户操控记录的生成时间来构建含有若干棵子树的用户操控动作森林

b)对用户操控动作森林进行遍历并生成事务集

c)通过改进的 FP-Growth 算法生成若干满足最小支持度的时序频繁项集

d)依据时序频繁项集形成满足最小置信度要求的时序候选关联规则

e)添加时间约束规则，进行时序候选关联规则筛选，通过规则的筛选的时序候选关联规则即用户满足时间约束为 q 分钟的时序关联操控习惯

2 用户操控动作森林与事务集的生成

用户的历史操控记录是按天存放的，如果简单将每一天的记录组成事务，将丢失用户在操控智能家居设备时所存在的部分关联关系，因此如何将用户历史操控数据转换为更加合理的事务集显得尤为重要。

针对用户操控记录，本文认为操作时间的间隔超过 30 分钟（忽略操作日期）的同一用户操控动作往往关联着不同的操控动作，若不加以区分就会产生错误的关联操控习惯。例如:若某一用户在早上 8 点喜欢打开烹饪机准备他的早餐，接着就会打开收音机听一下电台新闻早报;而到了晚上 6 点，该用户也会习惯性地打开烹饪机准备晚餐，而接着就是打开电视机，一边看着他喜欢的电视剧一边享用他的晚餐。在此背景下，针对该名用户应该存在两条关联规则：{早上 8 点:打开烹饪机→打开收音机}以及{晚上 6 点:打开烹饪机→打开电视机}，如果不对用户不同时刻的同一操控加以区分的话，就会产生关联操控习惯的漂移，即算法计算出打开烹饪机的平均操作时间是 13:00，在时间维度上远离早上 8 点附近打开电视机以及晚上 6 点附近打开电视机的操控动作，造成两个关联操控习惯的丢失。因此，需要将操作时间间隔已经超过 30 分钟的早上 8 点以及晚上 6 点打开烹饪机的操控动作认定为两个不同的用户操控动作，则能避免上述问题。

其次，为了高效地依据用户操控记录生成事务集，本文提出使用森林的方式存储用户操控记录，称为用户操控动作森林，并给出相应的森林的遍历方法以高效生成事务集，具体如下：

a)构建一个空的森林，并按照天遍历用户 n 天的历史操控记录，每一天生成一棵子树，最终形成含有 n 棵树的森林。

b)每天的子树的生成：构建一棵带有一个空节点的子树，然后按动作生成时间的先后顺序遍历用户操控记录的操控动作，并将该动作插入到根节点下从右往左第一个未包含该相同节点的分支的叶子节点后；若根节点下的所有分支均包含该动作，则将该操控动作存储为根节点的右孩子节点；

c)分别遍历森林里的动作子树，生成事务集：将森林中的各子树的根节点下每一个分支对应一个事务，最终多棵子树的多个分支所产生的多个事务组成针对该用户操控记录的事务集。

从理论角度出发，本文所提出的通过构建用户操控动作森林的方式来存储并生成事务集的时间复杂度优于传统的顺序表存储结构，证明如下：

假设现有某用户的 m 天所产生的 n 条用户记录：

如果使用传统的顺序表的存储方式，需要遍历顺序表的所有元素，如果该元素不存在在一构建的事务列表集合中则将其尾插到最近一个不同的动作所在的事务列表，否则将以其为表头新建一个事务列表。其时间复杂度为 $o(n^2)$ 级别[13]。

如果使用森林的方式生成事务集，则相当于构建含有 m 棵子树合计 n 个节点的用户操控动作森林的时间复杂度为 $o(n\log n)$，而遍历这个森林的时间复杂度为 $o(n\log n)$，合计时间复杂度为 $2o(n\log n)$，即 $o(n\log n)$ 级别[14]。因此，从理论角度可证明森林的处理方式在效率上由于传统的顺序表方式。其次，在后续的实验部分还将设定相应的验证实验以证

实猜想。

3　时序频繁项集的生成

在关联分析中，频繁项集代表的是一个规则出现的频率很高，即该规则频繁发生。通过计算规则的支持度可以体现出该规则在事务集中所占有的比例。定义规则{X ⟶ Y}的支持度(support)为事务集 D 中，同时包含用户操控动作 X 和 Y（记为动作 XY）的事务的百分比，即概率[15]。其中|*x*|代表包含用户操控动作 *x* 事务的个数，*len*(D)表示事务集中事务总数[15]。

$$\sup(XY) = \frac{|XY|}{len(D)} = P(XY) \qquad (1)$$

本算法提出基于 FP-Growth 算法进行频繁项集的生成。FP-Growth 算法生成频繁项集主要包括 FP 树的构建以及 FP 树的挖掘两大部分[16]。然而传统的 FP-Growth 算法在 FP-树的挖掘过程中仅利用支持度作为项头表的构建的依据，因此所产生的频繁项缺乏时序性，不满足智能家居推荐系统要求挖掘用户具有时间约束的时序关联操控规则的需求。为了解决上述问题，本文对 FP-树的挖掘过程进行了改进，通过添加时序化操作保证所生成的频繁项集满足时序的要求。

为了更加直观地说明频繁项集的生成过程，假设存在事务集:

T=

{{A,E,C,B },{ A,C,D,E },{ A,C,G },{ E,F,H },{ A,C,D,G},

{ A,C,E,G},{ A,E,F },{ B,G},{ A,C }}。

3.1　FP 树的构建

构建 FP 树主要包括项头表的构建、事务集的重构以及 FP 树的生成三大步骤[16]，具体流程如下：

a)项头表的构建。依据事务集的项头集 *I*={A,B,…}，计算对应的项头的支持度 $(\sup(x), x \in I)$，然后筛选出满足最小支持度(*min*(*supp*))要求的项头并对其进行排序后得到的该事务集的项头表，项头表形如:

{(D,*sup*(D)),((A,*sup*(A)),...}

且 $\min(\sup p) \le \sup(A) \le \sup(D)$

以假设的事务集 T 举例，针对事务 T 所生成的满足最小支持度为 10%的项表头如表 1 所示。

表 1　事务集 t 的项头表

Table 1　Headers table of transaction set T

事务集 T 的项头表				
事务 ID	事务详情	初始项头表	满足最小支持度的项头表	排序后的最终项头表
1	A,E,C,B	A:24%	A:24%	A:24%
2	A,C,D,E	B: 7%	C:21%	C:21%
3	A,C,G	C:21%	E:17%	E:17%
4	E,F,H	D: 7%	G:14%	G:14%
5	A,C,D,G	E:17%		
6	A,C,E,G	F: 7%		
7	A,E,F	G:14%		
8	B,G	H: 3%		
9	A,C			

b)事务集的重构。依据处理后的项头表，将各事务中不存在于项头表中的的元素进行删除，并对其按照项头表的顺序排序。以假设的事务 *T* 举例，重构后的事务集如表 2 所示。

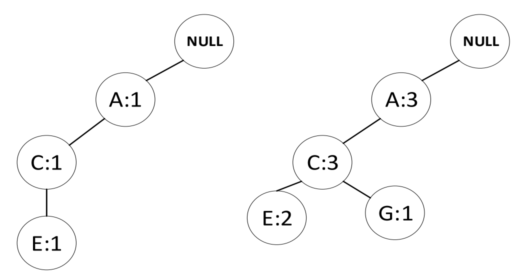
c)FP 树的生成。依据处理后的事务集按照其顺序进行树的层次插入。其中 FP 树的节点定义为 (x, x_n)，其中 x 为项头表的元素，n_x 为该项头到目前为止在该路径上出现的次数，即计数位。

表 2　事务集 t 的重构

Table 2　Reconstruction of transaction set T

事务 ID	事务详情	删除无效项头后的事务	排序后的事务
1	A,E,C,B	A,E,C	A,C,E
2	A,C,D,E	A,C,E	A,C,E
3	A,C,G	A,C,G	A,C,G
4	E,F,H	E	E
5	A,C,D,G	A,C,G	A,C,G
6	A,C,E,G	A,C,E,G	A,C,E,G
7	A,E,F	A,E	A,E
8	B,G	G	G
9	A,C	A,C	A,C

以假设的事务 *T* 举例，首先构建一个空的根节点，并将事务 1 插入到 FP 树中，生成如图 2(a)所示的 FP 树。接着将事务 2 插入到 FP 树中，生成如图 2(b)所示的 FP 树。依此类推，生成最终的 FP 树如图 3 所示。



(a)插入事务 1 后的 FP 树　　(b)插入事务 2 后的 FP 树

图 2　事务插入到 fp 树方式示意图

Fig. 2　Schematic diagram of inserting transaction to FP tree

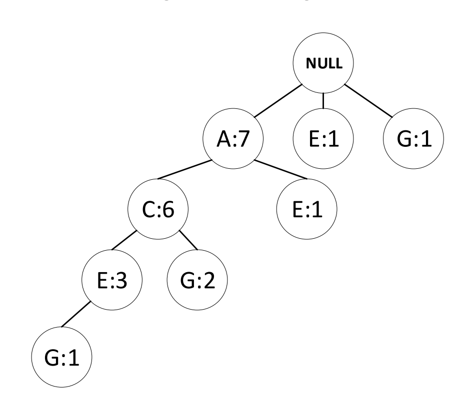


图 3　事务集 t 的 fp 树

Fig. 3　FP tree of transaction set T

3.2　FP 树的挖掘

通过对生成的 FP 树进行挖掘，可以得到相应的符合最小支持度要求的频繁项集。然而传统的 FP-Growth 算法在挖掘 FP 树过程中仅仅利用支持度作为处理的依据，无法产生时序的频繁项集，不能满足智能家具用户时序关联操控挖掘的需要。为了解决上述问题，本算法提出一种经过时序化处理改进的 FP 树挖掘算法，以挖掘出时序的频繁项集，具体步骤如下：

a)依照项头表的先后顺序，从后往前逆序遍历项头 x，

获得其对应的条件模式基：按 FP 树的层次结构，从根节点出发遍历到相应的项头节点，并更改所经过的节点的计数位 n_x。最后得到其相应的条件模式基，形式为

$$E_x = \{[(A,n_A),t_A],[(B,n_B),t_B],...,[(x,n_x),t_x]\}$$

其中 t_x 为元素（即用户操作动作）x 的平均操作时间（忽略操作日期）。

b)计算条件模式基的中除 x 的其他元素的支持度，并将不满足最小支持度要求的元素进行删除；

c)将处理后的条件模式基的元素进行时序化操作，最终形成时序条件模式基，其格式为：

$$E_x = \{[(A,n_A),t_A],[(B,n_B),t_B],...\}, t_A \le t_B$$

d)按照时序条件模式基的元素顺序进行组合，组合成若干个长度大于 1 包含 x 的频繁项。

e)迭代重复 a)b)，直到所有项头均被遍历为止为止，得到时序频繁项集

为了更好地说明生成频繁项集的过程，以事务集 T 锁所产生的 FP 树（图 3 所示）进行举例说明：

1)针对处于项头表最后一位的项头 G，产生其条件模式基：

$$E_G = \{[(A,3),t_A],[(C,3),t_C],[(E,1),t_E],[(G,3),t_G]\}$$

2)计算 E_G 中除去 G 后的各元素的支持度：

求解得到{(A:43%),(C43%),(E,14%)}，所有元素均满足最小支持度 10%的要求，均予以保留。

3)对经过筛选的条件模式基进行时序化排序：

假设 θ，则经过时序化处理的时序条件模式基为：

$$E_G = \{[(A,3),t_A],[(E,1),t_E],[(C,3),t_C],[(G,3),t_G]\}$$

4)针对时序条件模式基的元素按照原顺序进行组合得到针对该项头的长度大于 1 且包含 G 的时序频繁项集 P：

{(A:3, E:1, G:3), (A:3, C:3, G:3), (E:1, C:3, G:3), (A:3, E:1, C:3, G:3),}

5)如此类推，直至所有项头均被遍历，生成所有符合最小支持度要求的频繁项。

4　用户时序关联操控的产生

用户关联操控习惯的生成包括时序候选关联规则的生成以及通过对时序候选关联规则进行筛选后得到最终的用户操控习惯两大部分。首先，依据频繁项集生成满足最小置信度的时序候选关联规则，并计算候选关联规则时间约束因子，然后通过时间约束规则进行时序候选关联规则的筛选，挖掘出用户具有时间约束的时序关联操控习惯。

4.1　时序候选关联规则的生成

关联分析能在大量的数据中寻找数据之间的关系，这种关系以频繁项集或者关联规则两个形式存在。而在本算法中，为了更好地挖掘出符合用户兴趣爱好且可被用户理解的关联操控习惯，算法还必须通过计算各频繁项集的置信度来产生符合最小置信度要求的关联规则。

规则{X⟶Y}的置信度(confidence)是频事务集 T 中用户在执行动作 X 后直接或间接继续执行动作 Y 的事务（记为动作 Y|X）所占的百分比，即条件概率[17]为

$$conf(X \to Y) = \frac{\sup(XY)}{\sup(X)} = \frac{P(XY)}{P(X)} = P(Y \mid X) \quad (2)$$

以第 3.2 节针对项头 G 生成的频繁项集 P 为例：若某一频繁项 $P_i = \{A:3, E:1, G:3\}$，可以看出“动作 A，E，G 同时发生”的事件数受动作 E 影响为 1，而根据频繁项 θ 知“动作 A, E 同时发生”的事件数为 4，所以可以求得频繁项 P_i 的置信度为

$$conf(P_i) = \frac{P(AEG)}{P(AE)} = \frac{1}{4} = 0.25$$

如果将最小置信度设为 0.2，则可认为该频繁项所代表的时序关联规则{A⟶E⟶G}是有效的，则将其认定为时序候选关联规则，时序候选关联规则的格式形如：

$\{(A,t_A) \to (E,t_E) \to (G,t_G), \Delta t\}$，其中$\Delta t$为时间约束因子，代表该规则第二个规则子项的平均操控时间和第一个规则子项的平均操控时间的平均操控时间差，可由公式（3）求解：

$$\Delta t = t_{1,2} = t_2 - t_1 \quad (3)$$

4.2　时序候选关联规则的过滤

传统的关联规则分析算法在生成最终的关联规则的过程中仅利用置信度作为筛选标准，因此所产生的关联规则缺乏强烈的时间关联性，，为确保最终产生的关联规则的各规则子项之间存在强烈的时间关联性，本文提出采用一种基于动态时间约束因子Δt 的动态时间约束规则来进行时序候选关联规则筛选的方法，具体筛选过程如下：

a)每个时序候选关联规则都有一个动态因子，如果该规则的Δt 大于给定规则子项之间的时间约束系数 θ，则该规则被认为是无效的并被放弃。否则进行下一步筛选操作；

b)如果该规则不是无效的，则使用式(4)计算该规则中的第 i 个规则子项和第 j 个规则子项的平均时间差 $t_{i,j}$，假设该规则有 n 个规则子项：

$$t_{i,j} = t_j - t_i, 1 \le i < n-1, j = i+1 \quad (4)$$

c)如果 $t_{i,j} > \mathrm{D}t$，则在第 i 项后断开规则链并保留时序候选规则的前 i 项作为最终的规则，以确保规则具有强烈的时间关联性和时性特征。否则继续遍历下一对规则子项。

d)不断重复步骤 b)c)，直到该时序候选规则的所有条目都被遍历过为止。

e)对所有的时序候选关联规则重复步骤 a)~d)，直至所有时序候选规则均被遍历过为止。

时序候选关联规则的筛选算法流程描述如下：

输入：r:时序候选关联规则集

θ:规则子项之间的时间约束系数(单位:min)

输出：具有时间约束的时序关联规则

过程：

1. for 全部时序候选关联规则都被遍历过一次 do:
2. 　if Δt≤ θ then:
3. 　　for(i=1, j=2; j≤规则子项的个数;i++, j++) do:
4. 　　　　计算该规则中的第 i 个和第 j 个规则子项的平均时间差 $t_{i,j}$

```
        if t_{i,j} > Dt then
            在第 i 项后断开规则链; break
        end if
    end for
end if
if Δt> θ then
    认定该规则为无效规则并进行抛弃处理
end if
return 最终所有通过筛选的规则
```

5 实验与分析

5.1 实验环境

本文的验证实验的运行环境是一台具有 8 GB 内存且配有 Intel (R) Core^{TM} i7-6770 主频为 3.40 GHz 的 CPU 的个人计算机，该计算机运行 Windows 7 Professional 操作系统。本文的所有算法都是用 Python 编程语言编写的。

5.2 数据集

本节将利用由某智能家居公司提供的真实用户操控记录进行对比实验。经过删除包含缺失值的记录后，实验数据一共包含某用户近一年(2017 年 9 月 29 日到 2018 年 8 月 3 日）合计 557 372 条的操控记录，涉及 9 个智能家居设备。其次，根据公司的数据说明书，已知本文所使用的实验数据集一共含有 11 个具有较强关联性的时序关联操控习惯。实验数据的详情如表 3 所示。

表 3 实验数据

Table 3 Experimental data set

记录总数	557372
设备数量	9
记录开始时间	2017 年 9 月 29 日 15:46
记录结束时间	2018 年 8 月 3 日 15:22
设备详情	客厅窗帘、智能插座、客厅大灯、前门、后门、红外转发器、智能开关、风扇、空调
用户潜在的时序关联操控习惯	1.打开前门→打开客厅大灯→打开空调制冷模式 2.打开智能开关→打开智能插座→打开红外转发器 3.关闭空调→关闭客厅大灯→关闭前门 4.关闭客厅窗帘→打开大厅灯 5.关闭大厅灯 →打开客厅窗帘 6.关闭智能开关→关闭智能插座→关闭红外转发器 7.打开智能插座→打开大灯 8.打开空调制冷模式→打开空调除湿模式 9.关闭空调→打开风扇 10. 打开后门→关闭后门 11.关闭风扇→关打开空调制冷模式

5.3 实验结果与分析

为了本文所提出的事务集生成方式更高效，基于用户操控动作森林以及顺序表两种方式各进行了 50 次重复生成事务集的实验，两种事务集方法的每次实验的运行时间如图 4 所示。

最后，通过计算得出两种方法的生成事务集的平均运行时间分别为：用户操控森林方式平均耗时 79.80998 秒，传统顺序表方式生成同样的事务集则平均需要 120.41024 秒，效率提高了约 33.72%，实验结果证明本文所提出的基于用户操控动作树的事务集生成方法能更高效地生成关联规则分析阶段所需的事务集。

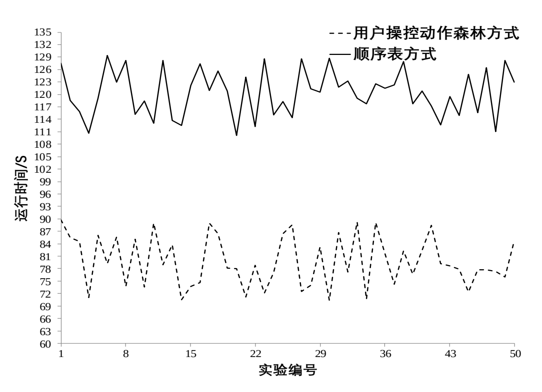


图 4 两种事务集生成方法的运行时间折线图

Fig. 4 Runtime comparison of two transaction set generation methods

其次，为了验证本文所提出算法能更加准确地挖掘出用户潜在的时序关联操控习惯，使用传统 Apriori 算法、传统 Eclat 算法、传统 FP-Growth 算法以及本文提出的 TAABFPG 算法进行重复实验，并使用 F1 值作为算法性能的最终评价标准，F1 值可通过式（5）求解[18]。

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5}$$

其中：*precision* 代表精准率，可根据式（6）求解；*recall* 代表召回率可根据式（7）求解。

$$precision = \frac{|\{relevent\} \cap \{retrieved\}|}{|\{retrieved\}|} \tag{6}$$

$$recall = \frac{|\{relevent\} \cap \{retrieved\}|}{|\{relevent\}|} \tag{7}$$

其中：$|\{relevent\}|$代表用户潜在的时序关联习惯个数；$|\{retrieved\}|$代表算法说挖掘到的规则（用户时序关联操控习惯）个数；$|\{relevent\} \cap \{retrieved\}|$代表算法说挖掘到的规则属于用户潜在的时序关联习惯的有效规则个数。

重复实验安排如下：设定最小支持度的取值范围为{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9}，最小置信度的取值范围同样为为{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9}；接着按顺序不重复随机组合最小支持度和最小置信度，共形成 49 对形如{0.6, 0.6}的{最小支持度，最小置信度}组合，然后分别使用 4 种算法进行 49 次设定有不同的最小支持度和最小置信度阈值的验证实验。

最后，针对验证实验的结果进行了统计与分析。表 4 展示了 4 种算法各自的每次实验输出的平均规则数、每次实验输出的平均有效规则数、平均精准率、平均召回率以及平均 F1 值。实验结果中的平均精准率、平均召回率以及平均 F1 值使用平均规则数以及平均有效规则数直接求解得出。

表 4 算法的各项指标(均值)

Table 4 Indicators comparision of four algorithms

算法	规则数	有效规则数	精准率	召回率	F1 值
Apriori	15.468	11.243	72.686%	93.69%	0.81863
ECLAT	15.462	11.241	72.701%	93.68%	0.81864
FP-Growth	15.469	11.245	72.694%	93.71%	0.81874
TAABFPG	13.449	11.375	84.579%	94.79%	0.89394

从表 4 可知，Apriori 算法、Eclat 算法以及 FP-Growth 算法的各项指标几乎一致，可以认定：若不考虑挖掘效率，三者的挖掘效果基本一样。然而本文提出的算法具有最高的精准率，说明本算法算法既能准确挖掘到潜在的正确规则又

不会产生过多的无用规则。其次，本文提出的算法具有最高的平均 F_1 值，说明本算法在挖掘用户时序关联操控习惯的性能上优于经典且常见的 Apriori 算法、Eclat 算法以及 FP-Growth 算法。因此，实验结果可以证实本文算法具有有效性并具有更好的性能，能更加准确地挖掘到用户潜在的具有时间约束的时序关联操控习惯。

6　结束语

为满足智能家居系统挖掘用户的具有时间约束的时序关联操控习惯的需求，本文基于 FP-Growth 算法，结合数据结构的知识，提出了一种能有效根据大量用户历史操控数据中挖掘用户时序关联操控习惯的关联分析算法。为提高事务集的生成效率，提出一种基于森林的数据结构存储用户操控记录，并通过遍历森林的方式高效生成事务集；其次，针对传统关联分析算法不能保证说挖掘到的关联规则的规则子项间存在时序性以及较强时间关联性的问题，提出一种有效的时间约束规则，该规则中通过计算动态时间约束因子来提高用户关联操控分析算法的精准率。最后，本文还选取了三种常见的关联分析算法作为基准算法做了多次重复实验，实验结果验证了本算法的 *n* 能大大提高事务集的生成效率，同时在挖掘用户时序关联操控习惯方面，有更优的性能。

其次，用户的操控习惯可能会受外部环境因素影响（如天气），而本文的验证实验中所使用的真实用户操控记录不包含外部环境因素的特征，无法利用外部环境因素指导用户操控习惯的挖掘。因此，如何收集到更多可能影响用户操控习惯发生显著性变化的外部环境因素，并将其融合到用户操控记录特征中，使得算法所挖掘到的用户操控习惯更加贴近用户真实情况，是下一步的研究内容。

参考文献：

[1] 童晓渝，房秉毅，张云．物联网智能家居发展分析 [J]. 移动通信，2010, 34(9): 16-20. (Tong Xiaoyu, Fang Bingyi, Zhang Yun. Analysis of the development of IoT smart home [J]. Mobile Communications, 2010, 34 (9): 16-20.)

[2] Wen Tianle. On the current situation and development tendency of smart home in China [C]// Proc of the 7th International Conference on Education, Sports, Arts and Management Engineering. Paris: Atlantis Press, 2017: 272.

[3] 朱敏玲，李宁．智能家居发展现状及未来浅析 [J]. 电视技术，2015, 39 (4): 82-85. (Zhu Minling, Li Ning. State of art and trend of smart home in China [J]. Video Engineering, 2015, 39 (4): 82-85.)

[4] Ji Enqing, Shi Haigang, Li Hongyi, *et al*. Research on new remote control platform for smart home system using mobile phones [C]// Proc of AMM. Switzerland: Trans Tech Publications, 2014: 267-274.

[5] 肖碧怡．面向智能家居的不确定性规则推理机制的研究与实现 [D]. 成都：电子科技大学, 2016. (Xiao Biyin. Design and implementation of rule based uncertainty reasoning for smart house [D]. Chengdu: University of Electronic and Technology of China, 2016.)

[6] Lee H S, Jung H W, Jung J Y, *et al*. A case study on the elderly people's behavior for developing smart home service-focus on analyzing behaviors filling up by oneself for 24hours- [J]. Journal of Materials Science Materials in Medicine, 2012, 23 (2): 307-14.

[7] Noury N, Hervé T, Rialle V, *et al*. Monitoring behavior in home using a smart fall sensor and position sensors [C]// Proc of the 1st Annual International Conference on Microtechnologies in Medicine and Biology. Piscataway, NJ: IEEE Press, 2000: 607-610.

[8] 吕培卓，戴洪涛．智能家居用户行为预测的方法研究 [J]. 中国新技术新产品, 2016 (3): 19-20. (Lyu Peizhuo, Dai Hongtai. Research on the method of predicting the behavior of smart home users [J]. China New Technology and New Products, 2016 (3): 19-20.)

[9] 孔英会，刘靖．智能家居中用户行为模式挖掘及控制策略研究 [J]. 电视技术，2013, 37(24): 39-42. (Kong Yinghui, Liu Jing. Research on data mining of user behavior and control strategy in smart home [J]. Video Engineering, 2013, 37 (24): 39-42.)

[10] Jin K K, Kim K B, Jo S. A service scenario generation scheme based on association rule mining for elderly surveillance system in a smart home environment [J]. Engineering Applications of Artificial Intelligence, 2012, 25 (7): 1355-1364.

[11] Heierman E O, Cook D J. Improving home automation by discovering regularly occurring device usage patterns [C]// Proc of the 3rd IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2003: 537.

[12] 牛邵峰．一种基于云端数据仓库的智能家居用户行为模式研究 [D]. 北京：北京邮电大学, 2014. (Niu Shaofeng. Research on the method of smart home pattern recognition based on data warehouse [D]. Beijing: Beijing University of Posts and Telecommunications, 2014.)

[13] 胡圣荣．数据结构教程与题解 [M]. 北京：清华大学出版社，2011: 16-30. (Hu Shengrong. Data structure tutorial and solution [M]. Beijing: Tsinghua University Press, 2011: 16-30.)

[14] 严蔚敏，吴伟民．数据结构 [M]. 北京：清华大学出版社，2009: 135-152. (Yan Weimin, Wu Weimin. Data structure [M]. Beijing: Tsinghua University Press, 2009: 135-152.)

[15] 周志华．机器学习 [M]. 北京：清华大学出版社，2016: 197-224. (Zhou Zhihua. Machine leaning [M]. Beijing: Tsinghua University Press, 2016: 197-224.)

[16] Li Haoyuan, Wang Yi, Zhang Dong, *et al*. Pfp: parallel FP-Growth for query recommendation [C]// Proc of ACM RECSYS. NewYork: ACM Press, 2008: 107-114.

[17] Harrington P. Machine Learning in Action [M]. Beijing: Post and Telecom Press, 2012: 200-243.

[18] 胡文江，胡大伟，高永兵，等. 基于关联规则与标签的好友推荐算法 [J]. 计算机工程与科学，2013, 35(2): 109-113. (Hu Wenjiang, Hu Dawei, Gao Yongbing, *et al*. Friend recommendation algorithm based on association rules and tags [J]. Computer Engineering and Science, 2013, 35 (2): 109-113.)